(2)

Robust Planning and Control Using Neural Networks

Grant N00014-89-J-3100

Summary of Technical Progress

June 30, 1990

W. Thomas Miller, III
Filson H. Glanz
L. Gordon Kraft, III
Michael J. Carter

Robotics Laboratory
Intelligent Structures Group
Department of Electrical and Computer Engineering
The University of New Hampshire

Contact: W. Thomas Miller, III
Department of Electrical and Computer Engineering
Kingsbury Hall
The University of New Hampshire
Durham, New Hampshire 03824
(603) 862-1326
t_miller@unhh.bitnet

Introduction

During the past five years, the Robotics Laboratory of the Department of Electrical and Computer Engineering at the University of New Hampshire has been studying the application of locally generalizing neural networks to difficult problems in control. In a series of theoretical and real time experimental studies, learning control approaches have been shown to be effective for controlling the dynamics of multidimensional, nonlinear robotic systems during repetitive and nonrepetitive operations. This project involves the extension of our work in learning control, with the combined goals of expanding our theoretical understanding of neural network based learning control systems and of extending our experimental work to include hierarchical learning control structures. Our work involves examining the efficacy of locally generalizing versus globally generalizing neural network architectures in control applications, as well as developing and analyzing learning control paradigms which are not restricted to specific network architectures. Various robotic systems within the laboratory form the basis for the real time experimental portions of the research. The concepts explored, however, should be applicable to a wide variety of control problems in addition to robotics.

In accordance with the above project goals, the ongoing work consists of five parallel efforts: system modeling, task planning, reinforcement learning, control system analysis, and fault tolerance. The project was approved for funding August 1, 1989, and work began September 1, 1989. ONR support was suspended temporarily in January, 1990, due to financial problems within DARPA. Work since that time has proceeded at a reduced level using funds preserved from the initial allocation. We have received some indication that funding will be resumed soon. This progress report summarizes activities during the period ending June 30, 1990.

A collection of recent publications has been included as an appendix.

System Modeling

Every control system incorporates some form of model of the system being controlled. Neural network learning appears to [illegible] well suited to model building in control since there is often a wealth of training data available from the [illegible]back sensors. The following efforts were proposed to extend our past work in neural network models for control:

1. Investigate neural network architectures which provide rapid performance convergence and which are resistant to learning interference during continuous on-line training in a control system.

2. Investigate non-recursive and recursive neural network architectures for modeling dynamical systems, including the effects of history dependence and time delays.

Research efforts during the current period have mostly been centered on the first of these two items. In particular, we have been examining alternative formulations of the CMAC neural network, as described in the December progress report. The traditional Albus CMAC network utilizes local receptive fields which are rectangular in shape and are distributed along hyper- diagonals in the network input hyperspace. We have been investigating CMAC neural networks with tapered, rather than rectangular, receptive fields. Such networks promise better (continuous) function approximation and well defined reverse derivatives. This work has focused on two issues: the shapes of the receptive fields and their placement in the multidimensional input space.

Our work with receptive field shape has emphasized multi-dimensional receptive fields formed from the outputs of overlapping one-dimensional receptive fields used to encode the individual input measurements. This is in contrast to typical radial basis functions, which form receptive fields using euclidean distances in the multidimensional space. While true radial basis functions have many nice mathematical properties, there seems to be better biological evidence for independently encoded sensors, and such decoupled sensing provides many advantages for distributed implementations. (Note that for gaussian receptive field shapes, the product of independent one-dimensional receptive fields for individual inputs is identical to a radial basis function based on euclidean distance. This is not true in general, however.) We have investigated different shapes for the independent receptive fields (rectangular, linear taper, gaussian taper) and different policies for formulating multi-dimensional tapered receptive fields from the fields of individual sensors (product of fields, minimum

strength of fields, etc.).

The issue of how to place the receptive fields in a multidimensional space is more difficult, and probably more important, than the issue of their exact shape. One strength of the CMAC neural network is derived from its use of locally generalizing receptive fields with field centers locations are fixed in location, regular in arrangement, and sparsely distributed. The fixed, regular arrangement is critical to the implementation of systems with very many receptive fields since it allows the efficient implementation of "virtual" fields (in hardware or software) with independent weights but common control structures. This is also important for rapid convergence, since it is often harder to adapt the placement of the fields (in a Kohonen network, for example) than to adjust the field strength once the placement has adapted. The sparseness of the distribution is important to assure that any single input does not excite too many receptive fields (an implementation issue). For low input dimensionality, fixed lattice arrangements can be found which are reasonably uniform. For higher dimensions, it is difficult to define sparse lattice arrangements with near uniform distributions. This problem is actively being pursued.

Task Planning

Our past work in neural network based learning control has emphasized the path following aspect of control, which involves successfully carrying out predetermined plans (e.g. robot arm trajectories). The following items of the work plan are intended to extend this research to include learned path planning, which is the next logical level of a control hierarchy.

1. Develop techniques for learned trajectory planning for systems with structural redundancy in the presence of obstacles. Evaluate the robustness of such learned planning systems for incompletely trained models.

2. Develop practical techniques for learning "optimal" trajectories for dynamical systems with constraints. Utility functions such as minimum time, minimum energy, minimum jerk, etc. will be investigated.

As discussed in the previous progress report, we have been studying sub-optimal trajectory planning for redundant systems using simple network training heurisitics, rather than mathematical optimization. We are currently extending these simulations to include obstacle avoidance issues. The results will then be compared with those obtained using more formal optimization techniques. Initial results indicate that such heuristic training procedures may reliably provide good solutions at relatively low learning effort, which may be adequate (and even desirable) in situations in which finding a true optimal solution is not necessary.

We plan to study the same planning issues in real time experiments, in order to assess the effects of realistic measurement uncertainties. The experimental platform to be utilized was largely developed during the current project period. It is based upon two desktop Scorbot-ER V articulated robot arms, each with five motion axes and a force sensing gripper. The twelve total servo axes (pulse width modulated DC motor drivers with optical position encoders) are controlled from a single 68010 based computer. The robots will be fitted with padded sleeves, allowing moderate speed contact with each other or with obstacles without damage and providing a crude localization of the point of contact. The experimental setup also includes a desktop Rhino XR-V articulated arm, modified in our laboratory to provide proportional motor control with optical encoder position feedback. The five major axes of this robot serve as the workspace transport for an active binocular vision system, allowing the vision system to be positioned dynamically in order to achieve the best task relative feedback. The robot has been fitted with a special hand which holds two small CCD video cameras with auto iris lenses. The original gripper axis of the robot has been modified to control the parallax angle between the two cameras. Vision system motion control and visual feature extraction is performed on a 33 MHz 80386 based computer with ITI FG100 imaging hardware. This imaging system includes a 1024x1024x12 frame buffer (configurable as multiple smaller frame buffers), dual analog video inputs, and limited real time processing using feedback look-up tables (real time difference/derivative images, etc). Most of this equipment was acquired using internal sources of funds (given the funding suspension, project support had to be reserved for graduate personnel costs).

Reinforcement Learning

We intend to study reinforcement learning within the context of learned biped walking. The immediate goal is to implement a control system which can learn to walk with dynamic balance, for a variety of slopes and payloads, using only crude models of the biped characteristics. The following investigations will proceed using a computer simulation and an experimental biped, both of which have been developed in our laboratory:

1. Develop a reinforcement learning architecture for adapting approximate walking trajectories precomputed using a crude biped model.

2. Evaluate and refine learning technique in the context of efficient biped walking on horizontal surfaces with variable payload.

3. Evaluate and refine learning technique in the context of efficient biped walking on sloped surfaces with variable payload.

Previous efforts in this section of the research were aimed at refining both the biped simulator and the physical model. Two different biped simulators were developed, one (discussed in the previous report) containing substantial detail and the second containing only first approximations to the dynamics. During the current project period, the simple simulator was used to investigate strategies for foot placement during dynamic walking. The objective was to develop a robust fixed strategy for foot placement using the simple model, and then to use neural network learning to adapt that strategy to accommodate differences between the simple model and the more complex simulator (and eventually, the physical biped). Experiments with the simple simulator have been completed, for a variety of walking conditions. Experiments using the more detailed simulator will proceed during the next project period.

Limited progress has been made with the experimental biped, partially due to the need to conserve funds. The current physical system was designed as a prototype for investigating geometries, motors, and sensors, but is not sufficiently rugged to withstand extensive walking experiments. A more rugged structure has been designed, but has not yet been implemented, due to a reluctance on our part to support the construction costs from the initial allocation. A three axis accelerometer was constructed as a balance sensor during the project period, however. Support software for this sensor was written and various calibration experiments performed.

Control System Analysis

Our past work has emphasized the use of neural networks as nonlinear models for adaptive control. The concepts have been demonstrated to be practical and effective for difficult simulated and real time experimental control problems. We plan to extend the theoretical analysis of these important learning control techniques in the following areas, using simplified system models:

1. Analyze learning control system performance in the presence of noisy measurements.

2. Develop stability criteria for the closed-loop learning control system in competing control architectures.

3. Analyze convergence times for the neural network weights, and settling times for the performance of the closed-loop learning control system.

4. Compare neural network based learning control with other adaptive control techniques.

Progress in these areas was documented in the previous report. Only limited new work in the area of system identification using neural networks was carried out during the current project period, due to other commitments by the faculty member involved.

Fault Tolerance

Fault tolerance is often mentioned as an attribute of artificial neural networks, although much of the current evidence is anecdotal in nature. Since fault tolerance is clearly an important feature for robust control, we plan to investigate neural network fault tolerance explicitly, as follows:

1. Operational fault tolerance. Study relationships between network size, degree of generalization, function complexity and function reproduction accuracy for networks with faults imposed after training. Using measures of complexity derived from information theory, develop unified bounds to the complexity of networks that will realize a function of given complexity, with a desired approximation accuracy, in the presence of faults.

2. Learning fault tolerance. Study relationships between learning convergence time, final approximation accuracy, and function complexity for faults imposed before training. Using the analogy between learning and system identification, extend known results on the identifiability of input-output systems to situations in which the identification system has parameter constraints (i.e. faults), and apply these to the determination of learning complexity.

3. Fault tolerance enhancement. Develop fault tolerant quantization schemes for a fixed input layer of the network, to create robust internal representations. Design internal representations based on results from diophantine approximation theory to retain representation accuracy in the presence of faults. Study techniques for "weight balancing" to minimize sensitivity to faults.

Previous work (reported in the prior progress report) involved the study of the fault tolerance of CMAC neural networks, with which our laboratory is primarily concerned. However, work in fault tolerance during the current project period focused on the fault tolerant aspects (operational fault tolerance) of multilayer perceptron networks. It was found that such networks are not inherently fault tolerant, in the sense that destroying a single weight in a network with 50-100 total weights (after training) can cause the RMS approximation error to exceed the RMS value of the learned function. This problem is typical for globally generalizing networks (as opposed to locally generalizing networks like CMAC, for which a single weight only effects the response in a limited region of the state space). An internal report is included in the Appendix. Further study is in progress, to determine the relationship between network size and fault tolerance and to investigate training algorithms which may improve fault tolerance.

Related Work in Progress

In January, 1990, we recently completed a very high speed *implementation of the CMAC neural network* using dedicated CMOS logic, rather than a general purpose or RISC processor (ONR grant N00014-89-J-1686). This technology was then used to implement two general purpose CMAC associative memory boards for the industry standard VME bus, facilitating future development of real time applications of neural networks to learning control systems, pattern recognition, and signal processing. Two prototype VME boards were constructed, each implementing a CMAC network with one million adjustable weights. VME bus response times for typical CMAC networks with 32 integer inputs and 8 integer outputs are on the order of 200 to 400 microseconds, depending on the network generalization parameter, making the networks sufficiently fast for most robot control problems, and many pattern recognition and signal processing problems. The two boards developed are being used by the Robotics Laboratory at the University of New Hampshire and by the Robot Systems Division of the National Institute of Standards and Technology. We recently designed a PC-AT bus version of this CMAC hardware and entered into a production agreement with the Shenandoah Systems Company in Newington, New Hampshire. Commercial versions of this hardware should be available in September, 1990.

As part of a NSF funded project, we have been developing design heuristics for the use of CMAC neural networks for modeling in control system applications. These heuristics are based on a compilation of experimental and simulation data obtained in our laboratory, showing the effects of various network design parameters on model accuracy and speed of training convergence. The goal is to provide a set of "rules" or "design criteria" which can be used by control system engineers with limited background in neural networks.

Bibliography

Miller, W. T., Hewes, R. P., Glanz, F. H., and Kraft, L. G., "Real Time Dynamic Control of an Industrial Manipulator Using a Neural Network Based Learning Controller." IEEE J. of Robotics and Automation, Vol. 6, pp. 1-9, 1990.

Miller, W. T., Box, B. A., and Whitney, E. C., "Design and Implementation of a High Speed CMAC Neural Network Using Logic Programmable CMOS Logic Cell Arrays." UNH Intelligent Structures Group Report ECE.IS.90.01, Feb. 6, 1990.

Miller, W. T., Box, B. A., Whitney, E. C., and Glynn, J. M., "Design and Implementation of a High Speed CMAC Neural Network Using Logic Programmable CMOS Logic Cell Arrays." To be presented at the Naval Underwater Systems Center Workshop on Neural Networks for Decision, Estimation, and Control, Newport, R.I., Aug. 7-8, 1990.

Miller, W. T., Box, B. A., Whitney, E. C., and Glynn, J. M., "Design and Implementation of a High Speed CMAC Neural Network Using Logic Programmable CMOS Logic Cell Arrays." Submitted to the IEEE Conference on Neural Information Processing Systems, Denver, CO, Nov. 26-29, 1990.

Miller, W. T., An, E., Glanz, F. H., and Carter, M. J., "The Design of CMAC Neural Networks for Control." To be presented at the Sixth Yale Workshop on Adaptive and Learning Systems, New Haven, CT, Aug. 15-17, 1990.

Miller, W. T., and Aldrich, C. M., "Rapid Learning Using CMAC Neural Networks: Real Time Control of an Unstable System." To be presented at the Fifth IEEE International Symposium on Intelligent Control, Phil., PA, Sept. 5-7, 1990.

Rudolph, F., "A Successful Robot Path Planner Using Supervised Training of a Direct Inverse Model with Excess Degrees of Freedom." Submitted to the IEEE Conference on Neural Information Processing Systems, Denver, CO, Nov. 26-29, 1990.

Segee, B. E., and Carter, M. J., "Fault Sensitivity and Nodal Relevance Relationships in Multi-layer Perceptrons." UNH Intelligent Structures Group Report ECE.IS.90.02, March 9, 1990.